

**LAPORAN PRAKTIKUM PEKAN 3**  
**ALGORITMA DAN PEMROGRAMAN**  
OPERATOR ARITMATIKA, PENUGASAN, RELASIONAL, DAN LOGIKA

Disusun oleh:

Thaariq Salam

2511532022

Dosen Pengampu: Dr. Wahyudi S.T.M.T

Asisten Praktikum: Rahmad Dwirizki Olders



**DEPARTEMEN INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS ANDALAS**  
**TAHUN 2025**

## KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas rahmat dan karunia-Nya sehingga Laporan Praktikum mata kuliah Pemrograman Algoritma dan Pemrograman dengan materi "Operator Aritmatika, Penugasan, Relasional, dan Logika" ini dapat diselesaikan dengan baik dan tepat waktu. Laporan ini disusun sebagai pemenuhan tugas praktikum Pekan 3 yang merupakan bagian integral dari proses pembelajaran di Departemen Informatika.

Laporan ini berfokus pada implementasi berbagai jenis operator dasar dalam bahasa pemrograman Java. Praktikum ini memberikan pengalaman langsung kepada penyusun untuk memahami bagaimana operator berfungsi sebagai pondasi logis dalam setiap alur eksekusi program, khususnya dalam konteks perhitungan, perbandingan, dan pengambilan keputusan.

Penyusun menyadari bahwa keberhasilan penyusunan laporan ini tidak lepas dari bimbingan dan dukungan berbagai pihak. Oleh karena itu, penyusun menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Bapak Dr. Wahyudi S.T.M.T selaku Dosen Pengampu mata kuliah Algoritma dan Pemrograman.
2. Rahmad Dwirizki Olders selaku Asisten Praktikum yang telah memberikan arahan teknis selama pelaksanaan praktikum.

Penyusun berharap laporan ini tidak hanya memenuhi aspek administratif, tetapi juga dapat menjadi dokumentasi yang bermanfaat dan media *review* bagi penyusun maupun pembaca lainnya. Kritik dan saran konstruktif sangat diharapkan demi perbaikan di masa mendatang.

Padang, 27 September 2025

Penulis

## DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI .....	ii
BAB I.....	1
PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Tujuan .....	1
1.3 Manfaat .....	2
BAB II.....	3
PEMBAHASAN .....	3
2.1 Pratikum Operator Algoritma.....	3
2.1.1 Kode program .....	3
2.1.2 Langkah kerja.....	4
2.1.3 Analisis hasil.....	4
2.2 Praktikum Operator Penugasan ( <i>Assignment</i> ) .....	5
2.2.1 Kode program .....	5
2.2.2 Langkah kerja.....	5
2.2.3 Analisis hasil .....	6
2.3 Praktikum Operator Logika .....	7
2.3.1 Kode program .....	7
2.3.2 Langkah kerja.....	7
2.3.3 Analisi hasil.....	8
2.4 Praktikum Operator Relasional.....	9
2.4.1 Kode program .....	9
2.4.2 Langkah kerja.....	9
2.4.3 Analisi hasil.....	10
BAB III .....	11
PENUTUP.....	11
3.1 Kesimpulan .....	11
DAFTAR PUSTAKA.....	12

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Praktikum pekan ini adalah *next level* dari pekan sebelumnya. Kalau sebelumnya kita berkenalan dengan tipe data dan variabel, sekarang kita belajar bagaimana data itu bisa kita *olah* dan kita *bandingkan*. Praktikum ini fokus ke materi Operator di bahasa pemrograman Java, yang merupakan tulang punggung dari setiap program.

Operator adalah simbol atau kata kunci yang berfungsi untuk melakukan manipulasi atau operasi tertentu pada satu atau lebih *operand* (nilai atau variabel). Bisa dikatakan, operator adalah *action word* dalam sebuah kode. Tanpa operator, program cuma bisa menyimpan data, tapi nggak bisa mikir, menghitung, atau mengambil keputusan.

Ada banyak jenis operator yang harus kita kuasai, mulai dari operator hitung-hitungan dasar (*Aritmatika*), cara cepat mengisi nilai (*Assignment*), sampai operator yang bikin program bisa 'berargumen' dan 'memutuskan' (*Relasional* dan *Logika*). Dengan mengimplementasikan semua operator ini secara langsung di lab, kami berharap bisa menutup *gap* antara teori di kelas dengan kenyataan *coding* di lapangan. Praktikum ini adalah fondasi yang wajib kuat sebelum masuk ke materi Struktur Kontrol yang lebih kompleks di pekan selanjutnya.

#### 1.2 Tujuan

Pelaksanaan praktikum Pekan 3 ini punya target utama agar mahasiswa bisa:

1. Mengimplementasikan Operator Aritmatika: Mampu menggunakan dan menganalisis hasil dari operator penjumlahan, pengurangan, perkalian,

pembagian, serta operator *modulo* (sisanya) yang penting dalam banyak algoritma.

2. Menguasai Operator Penugasan: Dapat menggunakan bentuk singkat dari operator penugasan ( +=, \*=) untuk membuat kode lebih efisien dan ringkas.
3. Memahami Operator Relasional: Mampu membandingkan dua nilai numerik menggunakan operator seperti > < == != dan mengerti bahwa hasilnya selalu berupa nilai *boolean* (*True* atau *False*).
4. Menerapkan Operator Logika: Mampu menggabungkan dua atau lebih kondisi *boolean* menggunakan operator Konjungsi (&&), Disjungsi (||), dan Negasi (!) yang sangat krusial dalam mekanisme pengambilan keputusan program.

### 1.3 Manfaat

Secara umum, praktikum mengenai operator ini memberikan manfaat fundamental yang signifikan bagi mahasiswa. Manfaat tersebut mencakup peningkatan kemampuan logika pemrograman karena mahasiswa dilatih untuk menyusun alur program dan kondisi secara algoritmik dan terstruktur. Selain itu, praktikum ini membiasakan mahasiswa untuk menulis kode yang lebih efisien dan ringkas melalui penggunaan operator penugasan ( +=, \*= dan lain-lain), yang merupakan standar praktik terbaik dalam pemrograman. Lebih lanjut, penguasaan operator relasional dan logika adalah bekal utama yang akan menjadi penentu alur kendali dalam materi percabangan dan perulangan di Java.

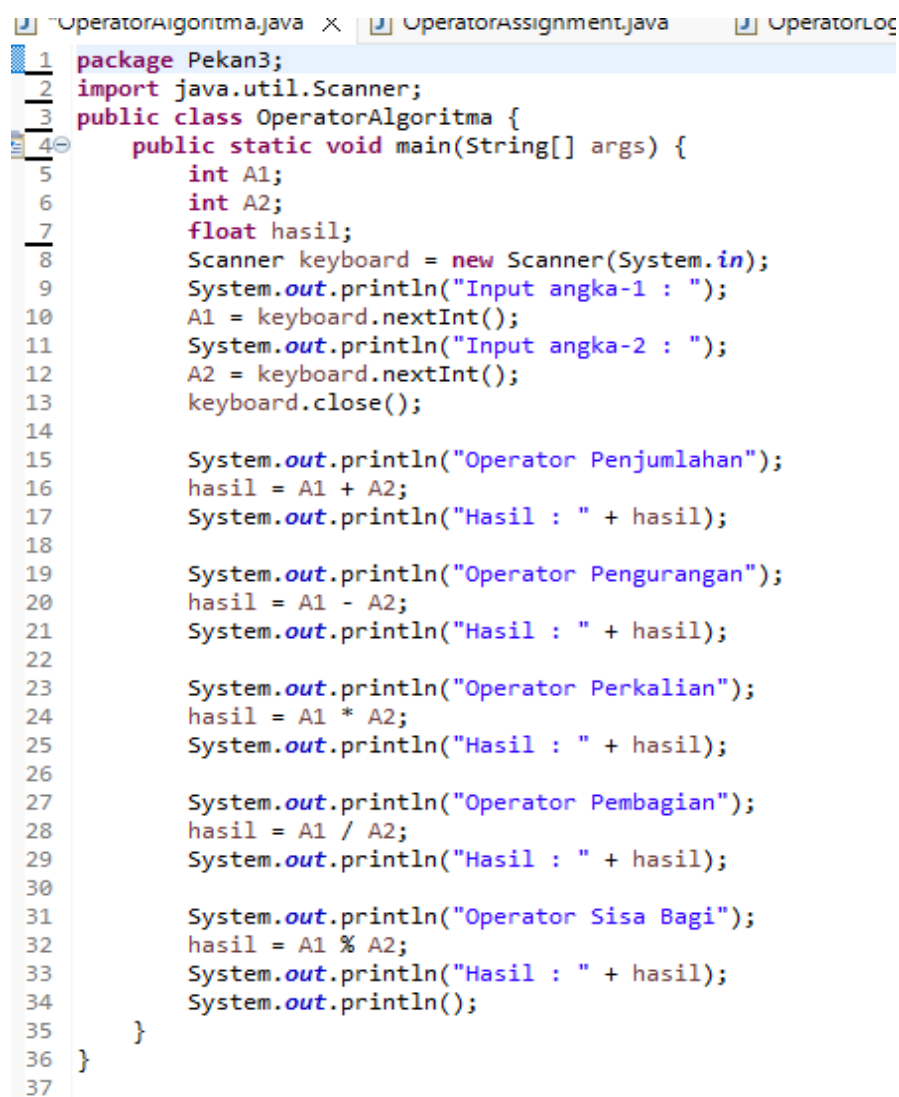
## BAB II

### PEMBAHASAN

#### 2.1 Pratikum Operator Algoritma

##### 2.1.1 Kode program

Kode program 2.1 menyajikan praktik penggunaan lima operator algoritma dasar yang berfungsi untuk melakukan perhitungan numerik. Operator Algoritma



```
1 package Pekan3;
2 import java.util.Scanner;
3 public class OperatorAlgoritma {
4     public static void main(String[] args) {
5         int A1;
6         int A2;
7         float hasil;
8         Scanner keyboard = new Scanner(System.in);
9         System.out.println("Input angka-1 : ");
10        A1 = keyboard.nextInt();
11        System.out.println("Input angka-2 : ");
12        A2 = keyboard.nextInt();
13        keyboard.close();
14
15        System.out.println("Operator Penjumlahan");
16        hasil = A1 + A2;
17        System.out.println("Hasil : " + hasil);
18
19        System.out.println("Operator Pengurangan");
20        hasil = A1 - A2;
21        System.out.println("Hasil : " + hasil);
22
23        System.out.println("Operator Perkalian");
24        hasil = A1 * A2;
25        System.out.println("Hasil : " + hasil);
26
27        System.out.println("Operator Pembagian");
28        hasil = A1 / A2;
29        System.out.println("Hasil : " + hasil);
30
31        System.out.println("Operator Sisa Bagi");
32        hasil = A1 % A2;
33        System.out.println("Hasil : " + hasil);
34        System.out.println();
35    }
36 }
37
```

Gambar 2.1

### 2.1.2 Langkah kerja

- 1) Buat berkas Java bernama `OperatorAlgoritma.java` di dalam *package* `Pekan3`.
- 2) *Import class* `Scanner` untuk memfasilitasi input data dari pengguna.
- 3) Deklarasikan variabel `A1` dan `A2` dengan tipe data `int`, serta variabel hasil dengan tipe data `float`.
- 4) Minta pengguna memasukkan dua nilai *integer* untuk `A1` dan `A2`.
- 5) Laksanakan operasi Penjumlahan, Pengurangan, Perkalian, Pembagian, dan Sisa Bagi (%) secara berurutan. Setiap hasil operasi ditugaskan ke variabel hasil.
- 6) Cetak hasil dari setiap operasi ke konsol.
- 7) Kompilasi dan eksekusi program.

### 2.1.3 Analisis hasil

Program ini berhasil mengimplementasikan lima operator aritmatika dasar. Fokus analisis terletak pada operasi Pembagian (/). Meskipun variabel hasil bertipe `float` (bilangan *floating-point*), *operand* `A1` dan `A2` bertipe `int`. Oleh karena itu, Java melaksanakan *integer division* (pembagian bilangan bulat) terlebih dahulu, yang secara otomatis menghilangkan atau membuang (*truncation*) bagian desimal dari hasil bagi. Hasil integer inilah yang kemudian dikonversi secara implisit menjadi *float* saat ditugaskan ke variabel hasil. Untuk mendapatkan hasil pembagian yang presisi, minimal salah satu *operand* harus diubah tipenya menjadi *float* atau *double* (misalnya dengan *casting* (`float`) `A1 / A2`).

Sementara itu, operator sisa bagi (%) berfungsi dengan efektif untuk mengembalikan sisa hasil bagi antara dua bilangan bulat, yang sering dimanfaatkan dalam algoritma untuk memeriksa bilangan genap/ganjil atau siklus waktu.

## 2.2 Praktikum Operator Penugasan (*Assignment*)

### 2.2.1 Kode program

```
1 package Pekan3;
2 import java.util.Scanner;
3 public class OperatorAssignment {
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         int A1;
7         int A2;
8
9         Scanner keyboard = new Scanner(System.in);
10
11         System.out.println("Inputkan Angka-1 : ");
12         A1 = keyboard.nextInt();
13         System.out.println("Inputkan Angka-2 : ");
14         A2 = keyboard.nextInt();
15
16         keyboard.close();
17
18         System.out.println("Assignment penambahan :");
19         A1 += A2;
20         System.out.println("Hasil : " + A1);
21
22         System.out.println("Assignment Pengurangan :");
23         A1 -= A2;
24         System.out.println("Hasil : " + A1);
25
26         System.out.println("Assignment Perkalian :");
27         A1 *= A2;
28         System.out.println("Hasil : " + A1);
29
30         System.out.println("Assignment Pembagian :");
31         A1 /= A2;
32         System.out.println("Hasil : " + A1);
33
34         System.out.println("Assignment Sisa Bagi :");
35         A1 %= A2;
36         System.out.println("Hasil : " + A2);
37     }
38 }
39
```

Gambar 2.2

### 2.2.2 Langkah kerja

- 1) Buat berkas Java bernama OperatorAssignment.java di dalam *package* Pekan3.



- 2) Deklarasikan variabel A1 dan A2 bertipe int dan terima input dari pengguna.
- 3) Laksanakan lima operasi penugasan gabungan: +=, -=, \*=, /=, dan %=.
- 4) Cetak hasil yang disimpan dalam variabel A1 setelah setiap operasi, karena operator ini memodifikasi nilai A1 secara langsung dan sekuensial.
- 5) Kompilasi dan eksekusi program.

### 2.2.3 Analisis hasil

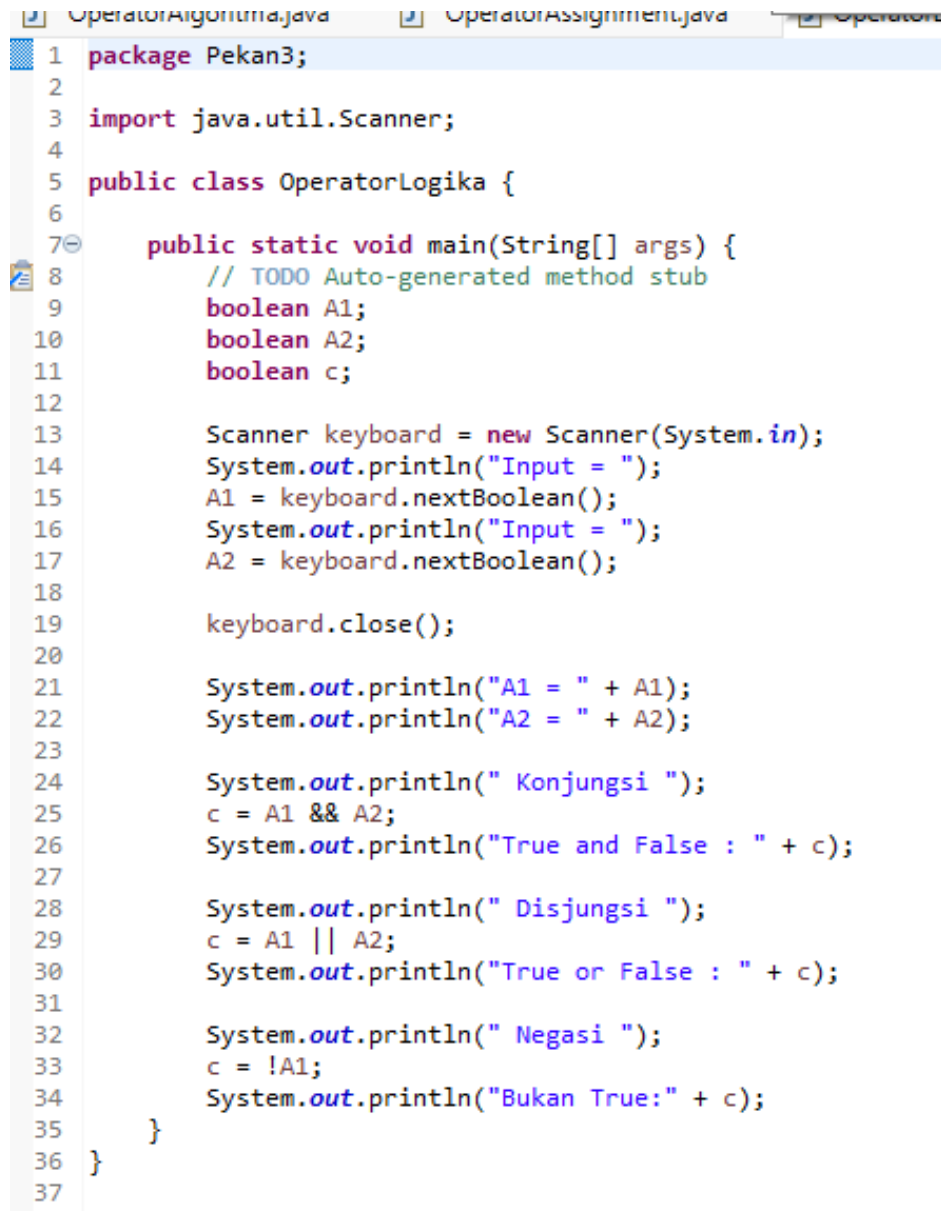
Operator penugasan gabungan (Compound Assignment Operators) berfungsi sebagai *shorthand* atau bentuk singkat dari penulisan ekspresi aritmatika lengkap. Sebagai contoh,  $A1 += A2$  secara fungsional setara dengan  $A1 = A1 + A2$ . Penggunaan operator ini bertujuan utama untuk efisiensi sintaksis dan membuat kode lebih mudah dibaca.

Hasil yang diperoleh menunjukkan adanya perubahan nilai sekuensial pada variabel A1. Setelah operasi pertama ( $A1 += A2$ ), nilai A1 yang baru akan digunakan sebagai *operand* pertama untuk operasi berikutnya ( $A1 -= A2$ ), dan seterusnya. Hal ini berarti setiap hasil operasi bersifat kumulatif dan bergantung pada hasil sebelumnya.

## 2.3 Praktikum Operator Logika

### 2.3.1 Kode program

Kode program 2.3 digunakan untuk mengimplementasikan operasi logika dasar pada tipe data *boolean* yang berfungsi sebagai penentu kondisi.



```
1 package Pekan3;
2
3 import java.util.Scanner;
4
5 public class OperatorLogika {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         boolean A1;
10        boolean A2;
11        boolean c;
12
13        Scanner keyboard = new Scanner(System.in);
14        System.out.println("Input = ");
15        A1 = keyboard.nextBoolean();
16        System.out.println("Input = ");
17        A2 = keyboard.nextBoolean();
18
19        keyboard.close();
20
21        System.out.println("A1 = " + A1);
22        System.out.println("A2 = " + A2);
23
24        System.out.println(" Konjungsi ");
25        c = A1 && A2;
26        System.out.println("True and False : " + c);
27
28        System.out.println(" Disjungsi ");
29        c = A1 || A2;
30        System.out.println("True or False : " + c);
31
32        System.out.println(" Negasi ");
33        c = !A1;
34        System.out.println("Bukan True:" + c);
35    }
36 }
37
```

Gambar 2.3

### 2.3.2 Langkah kerja

- 1) Buat berkas Java bernama OperatorLogika.java di dalam *package* Pekan3.
- 2) Deklarasikan variabel A1, A2, dan c sebagai *boolean*.

- 3) Gunakan *Scanner* untuk meminta pengguna memasukkan nilai *true* atau *false* untuk A1 dan A2.
- 4) Laksanakan dan cetak hasil dari tiga operator logika:

Konjungsi (&& - AND), Disjungsi (|| - OR), Negasi (! - NOT)

- 5) Kompilasi dan eksekusi program.

### 2.3.3 Analisi hasil

Operator Logika adalah elemen inti dalam menentukan aliran kendali program. Operator ini hanya bekerja pada *operand* bertipe *boolean* dan selalu menghasilkan nilai *boolean* sebagai keluarannya.

1. Konjungsi (&&): Operator AND. Prinsipnya adalah *true* hanya akan dihasilkan jika kedua *operand* (A1 dan A2) bernilai *true*. Jika salah satu bernilai *false*, hasilnya pasti *false*. Ini sering disebut sebagai *short-circuit AND*, karena evaluasi akan berhenti dan langsung menghasilkan *false* jika *operand* pertama sudah *false*.
2. Disjungsi (||): Operator OR. Prinsipnya adalah *true* akan dihasilkan jika setidaknya satu *operand* bernilai *true*. Hasilnya hanya akan *false* jika kedua *operand* bernilai *false*. Ini juga merupakan *short-circuit OR*, di mana evaluasi akan berhenti jika *operand* pertama sudah *true*.
3. Negasi (!): Operator NOT. Operator ini berfungsi untuk membalikkan nilai logika dari *operand* tunggal. Jika A1 adalah *true*, maka !A1 adalah *false*, dan sebaliknya.

Pemahaman terhadap tabel kebenaran dari operator ini sangat vital sebagai dasar penulisan ekspresi bersyarat dalam pernyataan *if*, *while*, atau *for*.

## 2.4 Praktikum Operator Relasional

### 2.4.1 Kode program

```
1 package Pekan3;
2 import java.util.Scanner;
3 public class OperatorRelasional {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8         int A1;
9         int A2;
10        boolean hasil;
11
12        Scanner keyboard = new Scanner(System.in);
13
14        System.out.println("Masukan Angka-1 :");
15        A1 = keyboard.nextInt();
16        System.out.println("Masukan Angka-2 :");
17        A2 = keyboard.nextInt();
18
19        keyboard.close();
20
21        hasil = A1 > A2;
22        System.out.println("Apakah A1 > A2 : " + hasil);
23        hasil = A1 < A2;
24        System.out.println("Apakah A1 < A2 : " + hasil);
25        hasil = A1 >= A2;
26        System.out.println("Apakah A1 >= A2 : " + hasil);
27        hasil = A1 <= A2;
28        System.out.println("Apakah A1 <= A2 : " + hasil);
29        hasil = A1 == A2;
30        System.out.println("Apakah A1 == A2 : " + hasil);
31        hasil = A1 != A2;
32        System.out.println("Apakah A1 != A2 : " + hasil);
33    }
34 }
35
```

Gambar 2.4

### 2.4.2 Langkah kerja

- 1) Buat berkas Java bernama OperatorRelasional.java di dalam *package* Pekan3.
- 2) Deklarasikan variabel A1 dan A2 sebagai int, serta hasil sebagai boolean.
- 3) Gunakan *Scanner* untuk menerima dua nilai *integer* dari pengguna.
- 4) Laksanakan dan cetak hasil dari keenam perbandingan relasional secara berurutan: lebih dari (>), kurang dari (<), lebih dari sama dengan (>=),

kurang dari sama dengan (<=), sama dengan (==), dan tidak sama dengan (!=).

- 5) Setiap hasil perbandingan ditugaskan ke variabel *boolean* hasil.
- 6) Kompilasi dan eksekusi program.

#### 2.4.3 Analisi hasil

Operator relasional digunakan untuk menentukan hubungan perbandingan antara dua nilai. Karakteristik utama dari operator ini adalah keluarannya yang selalu berupa nilai *boolean* (*true* atau *false*), yang mengindikasikan apakah hubungan yang dinyatakan itu benar atau salah.

1. Perbandingan Ukuran: Operator seperti >, <, >=, dan <= membandingkan besar kecilnya nilai.
2. Operator Kesamaan: Operator == digunakan untuk membandingkan apakah nilai kedua *operand* sama persis. Perlu diingat bahwa operator ini berbeda dengan operator penugasan (=).
3. Operator Ketidaksamaan: Operator != digunakan untuk memeriksa apakah nilai kedua *operand* berbeda.

Operator relasional membentuk kondisi dasar bagi pernyataan *if-else* dan perulangan *while/for*. Setiap pernyataan bersyarat dalam Java harus dimulai dengan evaluasi relasional untuk memutuskan alur eksekusi program selanjutnya.

## BAB III

### PENUTUP

#### 3.1 Kesimpulan

Praktikum Pekan 3 mengenai operator di Java telah berhasil dilaksanakan. Pengimplementasian empat jenis operator utama (Aritmatika, Penugasan, Logika, dan Relasional) memperkuat pemahaman mengenai mekanisme dasar pengambilan keputusan dan pemrosesan data dalam pemrograman. Poin-poin kesimpulan yang dapat ditarik adalah:

1. Operator Aritmatika berfungsi untuk operasi hitungan dasar. Perhatian khusus diperlukan pada operasi pembagian (/) antara *integer* (Integer Division) karena dapat menyebabkan hilangnya presisi desimal (*truncation*).
2. Operator Penugasan Gabungan (+=, \*= dan lain-lain) berfungsi mengefisienkan penulisan kode dengan menggabungkan operasi aritmatika dan penugasan. Penggunaannya harus memperhatikan perubahan nilai variabel secara sekuensial dan kumulatif.
3. Operator Relasional (<, ==, !=, dsb.) berperan penting dalam membandingkan *operand* dan selalu menghasilkan nilai *boolean* (true atau false), yang menjadi dasar utama bagi semua struktur kendali (*control structure*).
4. Operator Logika (&&, ||, !) digunakan untuk mengevaluasi dan menggabungkan kondisi *boolean*. Operator ini sangat krusial dalam membentuk ekspresi bersyarat yang kompleks untuk mengendalikan alur eksekusi program.

## DAFTAR PUSTAKA

- [1] A. Silberschatz, P. B. Galvin, dan G. Gagne, *Operating System Concepts*, 9th ed. Hoboken, NJ: Wiley, 2012.
- [2] Oracle, "The Java Tutorials," 2023. [Daring]. Tersedia pada: <https://docs.oracle.com/javase/tutorial/>. [Diakses: [27 September 2025]].